

Amendments to the Claims:

This listing of claims replaces all prior listings of claims:

1-19 (Canceled).

20. (Previously Presented) A system for synchronizing data objects for a user between a primary platform and a plurality of auxiliary platforms comprising:

a memory; and

processing means, coupled to the memory, to execute at least one computer program for:

creating, by a primary platform synchronization framework, a set of generic messages identifying changes to the data objects on the primary platform since a previous synchronization, the generic message not being dependent on a specific platform;

accessing a database to obtain a user identifier, the user identifier being associated with the user and linking to two or more device identifiers, the device identifiers identifying the plurality of auxiliary platforms;

converting, by a primary platform synchronization adapter, the generic messages to adapted messages corresponding to each of the auxiliary platforms based on the obtained user identifier and the linked device identifiers, the adapted messages being in adapted message formats used by underlying synchronization software;

sending the adapted messages from the primary platform to auxiliary platform synchronization adapters in the corresponding auxiliary platforms;
converting, by the corresponding auxiliary platform synchronization adapters, the adapted messages to generic messages on each of the auxiliary platforms; and
executing, by an auxiliary synchronization framework on each corresponding auxiliary platform, add, modify, and/or delete functions in the generic messages to synchronize the data objects used by a corresponding auxiliary platform application.

21. (Previously Presented) A computer-implemented method for synchronizing data objects for a user between a primary platform and a plurality of auxiliary platforms comprising

accessing a synchronization store database to obtain data characterizing changes to the data objects on the primary platform since a previous synchronization;

creating a set of generic messages identifying changes to the data objects on the primary platform since the previous synchronization based on the data obtained from the synchronization store database, the generic messages being platform independent;

accessing a synchronization database, the database comprising a first table with entries linking a user identifier for the user with two or more device identifiers stored in the database, the device identifiers identifying the plurality of auxiliary platforms, the adapted messages being in a form compatible with underlying synchronization software executing on the corresponding auxiliary platform;

converting the generic messages to adapted messages corresponding to each of the auxiliary platforms based on the linked device identifiers;

sending the adapted messages from the primary platform to the corresponding auxiliary platforms;

converting the adapted messages to generic messages on each of the auxiliary platforms; and

executing the generic messages on the corresponding auxiliary platform to update the data objects on the corresponding auxiliary platform.

22. (Previously Presented) A method as in claim 21, wherein all data objects supporting a particular transaction are grouped together in the synchronization store database, the method further comprising:

determining that an error is generated with respect to one or more data objects in a group associated with the transaction; and

canceling all data objects synchronizations relating to the transaction.

23. (Previously Presented) A method as in claim 21, wherein all data objects supporting a particular transaction are grouped together in the synchronization store database, the method further comprising:

determining that an error is generated with respect to one or more data objects in a group associated with the transaction; and

rolling back all data objects synchronizations relating to the transaction.

24. (Currently Amended) A system comprising:

a principal computing platform comprising memory storing instructions forming an operating system, an application program having one or more data objects for synchronization, a synchronization framework module for providing generic synchronization functions, and a synchronization adapter for communication messages and data objects between the synchronization framework module and synchronization software for providing underlying synchronization software, the synchronization framework module lying between the application program and the synchronization adapter;

an auxiliary computing platform exchanging messages with the principal computing platform over a communications link;

a synchronization database in communication with the principal communication platform storing data relating to users and one or more of their corresponding auxiliary computing platforms; and

a synchronization store database in communication with the principal communication platform storing data relating to data objects to be synchronized;

the synchronization module comprising:

an application interface module to transfer data objects to and from the principal computing platform;

a settings module to permit a user to select various settings and parameters for synchronization;

a selection module to select appropriate data objects to pass from principal computing platform to synchronization store database during synchronization;

a synchronization store module to create a copy of data objects received from principal computing platform via the selection module;

a synchronization engine module to retrieve content of the synchronizations store to be synchronized;

query builder module interfacing with the synchronization engine module to build generic messages to pass to synchronization adapter; and

an inbound queue module to retrieve[[s]] inbound messages from the auxiliary computing platform.

25. (New) A system as in claim 24, wherein the synchronization store module groups, for each of a plurality of transactions, related data objects within the synchronization database.

26. (New) A system as in claim 25, wherein all data objects relating to a particular transaction are canceled in response to an error associated with the transaction while the data objects that are not associated with the transaction are not canceled.

27. (New) A system as in claim 25, wherein all data objects relating to a particular transaction are rolled back in response to an error associated with the transaction while the data objects that are not associated with the transaction are not rolled back.

28. (New) A computer-implemented method comprising:

retrieving data objects to be synchronized from a primary computing platform application;

storing the fetched data objects;

comparing the stored fetched data objects to generate a delta set of changes by:

determining whether the fetched data objects are associated with a previously selected delta generation process;

using the previously selected delta generation process to compare the fetched data objects to a replica data set if it has been determined that the fetched data objects are associated with the previously selected delta generation process; or

using a default delta generation process to compare the fetched data objects to the replica data if it has not been determined that the fetched data objects are associated with the previously selected delta generation process;

building generic messages to implement the generated delta set of changes; and

changing the generic messages to adapted messages, the adapted messages being in a form compatible with underlying synchronization software; and

sending the adapted messages to an auxiliary computing platform.